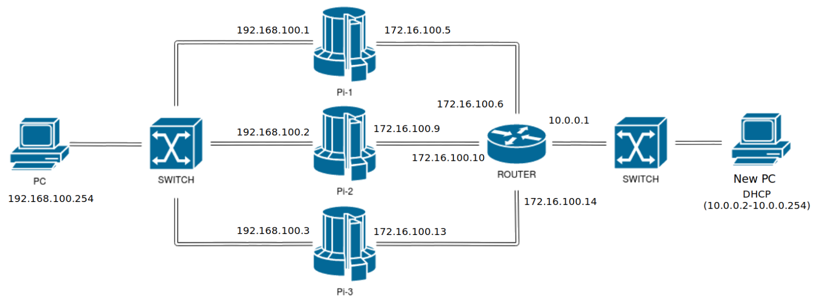
Systems and Devices 2 (Network)

Lec 4b: Network Layer

Before we get started ...

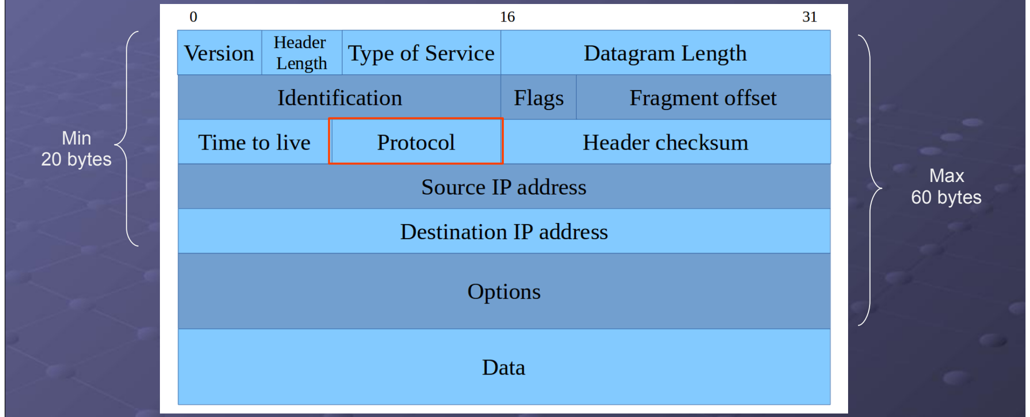
- How did your research go into:
  - Time To Live (TTL)
  - Fragmentation?
- We have looked at how packets are transferred at the network layer i.e. host to host, but how do routers know where to send each packet ?
  - Could use static rules, but these would need to be managed manually and in an ever changing world that is a lot of work. Therefore, we need to automate this process by using a protocol:
    - RIP, RIP2, OSPF, BGP ...

Quick Quizzz



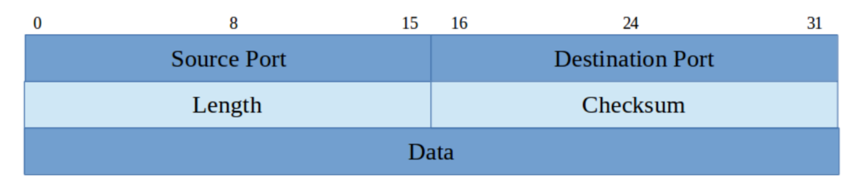
- What would we need to do to assign the “New PC” an IP address and then allow it to connect to PI-2 using address 192.168.100.2?
  - Hint, what allocates IP addresses & how are they routed?

IPv4



- In addition logical addressing the IP protocol is used to transport other protocols across a network.

Fragmentation

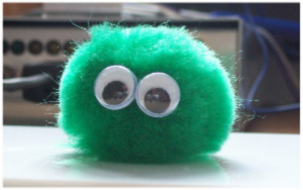


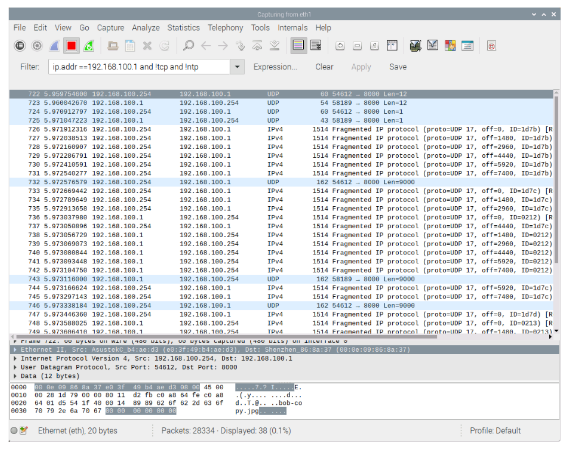
Source Port	Destination Port
Length	Checksum
Data	

MTU	IP header	TCP header	Data	Efficiency	Type
576	20	20	536	93.1%	Min
1500	20	20	1460	97.3%	Normal
9000	20	20	8960	99.5%	Jumbo
64000	20	20	63960	99.9%	Max

- Mismatch between transport and network layers
  - Maximum Transmission Unit (MTU)

Demo





Filter: ip.addr ==192.168.100.1 and !tcp and !ntp

```
722 5.959754600 192.168.100.254    192.168.100.1      UDP     60 54612 → 8000 Len=12
723 5.960042670 192.168.100.1      192.168.100.254    UDP     54 58189 → 8000 Len=12
724 5.970912797 192.168.100.254    192.168.100.1      UDP     60 54612 → 8000 Len=1
725 5.971047223 192.168.100.1      192.168.100.254    UDP     43 58189 → 8000 Len=1
726 5.971912316 192.168.100.254    192.168.100.1      IPv4  1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=1d7b) [R
727 5.972038513 192.168.100.254    192.168.100.1      IPv4  1514 Fragmented IP protocol (proto=UDP 17, off=1480, ID=1d7b)
728 5.972160907 192.168.100.254    192.168.100.1      IPv4  1514 Fragmented IP protocol (proto=UDP 17, off=2960, ID=1d7b)
729 5.972286791 192.168.100.254    192.168.100.1      IPv4  1514 Fragmented IP protocol (proto=UDP 17, off=4440, ID=1d7b)
730 5.972410591 192.168.100.254    192.168.100.1      IPv4  1514 Fragmented IP protocol (proto=UDP 17, off=5920, ID=1d7b)
731 5.972540277 192.168.100.254    192.168.100.1      IPv4  1514 Fragmented IP protocol (proto=UDP 17, off=7400, ID=1d7b)
732 5.972576579 192.168.100.254    192.168.100.1      UDP    162 54612 → 8000 Len=9000
733 5.972669442 192.168.100.254    192.168.100.1      IPv4  1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=1d7c) [R
734 5.972789649 192.168.100.254    192.168.100.1      IPv4  1514 Fragmented IP protocol (proto=UDP 17, off=1480, ID=1d7c)
735 5.972913658 192.168.100.254    192.168.100.1      IPv4  1514 Fragmented IP protocol (proto=UDP 17, off=2960, ID=1d7c)
736 5.973037980 192.168.100.1      192.168.100.254    IPv4  1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=0212) [R
737 5.973050896 192.168.100.254    192.168.100.1      IPv4  1514 Fragmented IP protocol (proto=UDP 17, off=4440, ID=1d7c)
738 5.973056729 192.168.100.1      192.168.100.254    IPv4  1514 Fragmented IP protocol (proto=UDP 17, off=1480, ID=0212)
739 5.973069073 192.168.100.1      192.168.100.254    IPv4  1514 Fragmented IP protocol (proto=UDP 17, off=2960, ID=0212)
740 5.973080844 192.168.100.1      192.168.100.254    IPv4  1514 Fragmented IP protocol (proto=UDP 17, off=4440, ID=0212)
741 5.973093448 192.168.100.1      192.168.100.254    IPv4  1514 Fragmented IP protocol (proto=UDP 17, off=5920, ID=0212)
742 5.973104750 192.168.100.1      192.168.100.254    IPv4  1514 Fragmented IP protocol (proto=UDP 17, off=7400, ID=0212)
743 5.973116000 192.168.100.1      192.168.100.254    UDP    162 58189 → 8000 Len=9000
744 5.973166624 192.168.100.254    192.168.100.1      IPv4  1514 Fragmented IP protocol (proto=UDP 17, off=5920, ID=1d7c)
745 5.973297143 192.168.100.254    192.168.100.1      IPv4  1514 Fragmented IP protocol (proto=UDP 17, off=7400, ID=1d7c)
746 5.973338184 192.168.100.254    192.168.100.1      UDP    162 54612 → 8000 Len=9000
747 5.973446360 192.168.100.254    192.168.100.1      IPv4  1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=1d7d) [R
748 5.973588025 192.168.100.1      192.168.100.254    IPv4  1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=0213) [R
749 5.973606410 192.168.100.1      192.168.100.254    IPv4  1514 Fragmented IP protocol (proto=UDP 17, off=1480, ID=0213)
```

```
▸ Frame 722: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▸ Ethernet II, Src: AsustekC_b4:ae:d3 (e0:3f:49:b4:ae:d3), Dst: Shenzhen_86:8a:37 (00:0e:09:86:8a:37)
▸ Internet Protocol Version 4, Src: 192.168.100.254, Dst: 192.168.100.1
▸ User Datagram Protocol, Src Port: 54612, Dst Port: 8000
▸ Data (12 bytes)

0000  00 0e 09 86 8a 37 e0 3f  49 b4 ae d3 08 00 45 00   .....7.? I.....E.
0010  00 28 1d 79 00 00 80 11  d2 fb c0 a8 64 fe c0 a8   .(.y.... ....d...
0020  64 01 d5 54 1f 40 00 14  89 89 62 6f 62 2d 63 6f   d..T.@.. ..bob-co
0030  70 79 2e 6a 70 67 00 00  00 00 00 00               py.jpg.. ....
```

Ethernet (eth), 20 bytes · Packets: 28334 · Displayed: 38 (0.1%) · Profile: Default

- What happens when we update the UDP file transfer program to use “Jumbo” packets.

TTL

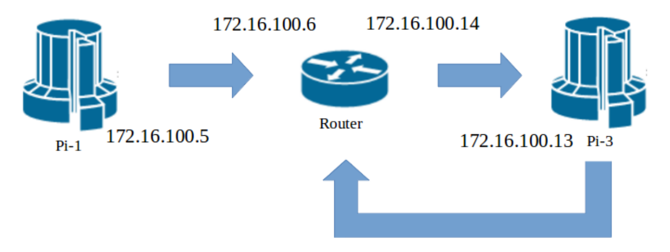
- To remove lost packets from a network we use the Time To Live (TTL) field
  - Typical values 64, 128, 255
  - TTL field decremented each time a packet is passed through a router. If 0 after subtraction dropped by router
  - Normal occurs when a network loop is accidentally created.
- Also used to restrict what networks the packet will be visible on (approximate).
  - 1 : local network (decremented when forwarded)
  - 4 - 8 : campus
  - 16 - 64 : world
  - 255 : unrestricted

Demo

ping pi-2	WORKS
ping pi-2.desk85.lan	WORKS
ping -c 2 -t 1 172.16.85.6	WORKS
ping -c 2 -t 1 172.16.85.9	DOES NOT WORK
ping -s 1500 pi-2.desk85.lan	WORKS
ping -s 1500 -M do pi-2.desk85.lan	DOES NOT WORK

- Some interesting PING commands
  - Q : if you were using DESK 85, why do some of these PING commands work and others do not?

Quick Quizzz



- Q : examine the routing tables on the next slide for Pi-1, Router and Pi-3. Can you see why pinging the IP address 10.200.0.1 from Pi-1 will cause a network loop?

Quick Quizzz

- Pi-1

```
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         172.16.100.6    0.0.0.0         UG    0      0        0 eth0
10.100.1.0      0.0.0.0         255.255.255.0   U     0      0        0 tun0
10.100.2.0      0.0.0.0         255.255.255.0   U     0      0        0 tun0
10.100.3.0      0.0.0.0         255.255.255.0   U     0      0        0 tun0
172.16.100.4    0.0.0.0         255.255.255.252 U     0      0        0 eth0
192.168.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
```

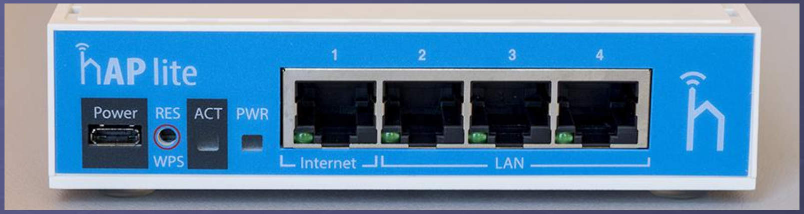
- Router

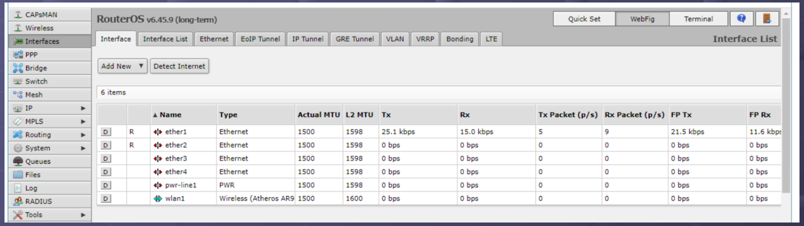
```
 #      DST-ADDRESS        PREF-SRC        GATEWAY            DISTANCE
 0 A S  10.200.0.0/24                      172.16.100.13             1
 1 ADC  172.16.100.4/30    172.16.100.6    ether1                    0
 2 ADC  172.16.100.8/30    172.16.100.10   ether2                    0
 3 ADC  172.16.100.12/30   172.16.100.14   ether3                    0
 4  DC  172.16.100.16/30   172.16.100.18   ether4                  255
```

- Pi-3

```
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         172.16.100.14   0.0.0.0         UG    0      0        0 eth0
10.100.1.0      0.0.0.0         255.255.255.0   U     0      0        0 tun0
10.100.2.0      0.0.0.0         255.255.255.0   U     0      0        0 tun0
10.100.3.0      0.0.0.0         255.255.255.0   U     0      0        0 tun0
172.16.100.12   0.0.0.0         255.255.255.252 U     0      0        0 eth0
192.168.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
```

Demo





RouterOS v6.45.9 (long-term)

Interface List

6 items

		▲ Name	Type	Actual MTU	L2 MTU	Tx	Rx	Tx Packet (p/s)	Rx Packet (p/s)	FP Tx	FP Rx
D	R	ether1	Ethernet	1500	1598	25.1 kbps	15.0 kbps	5	9	21.5 kbps	11.6 kbp
D	R	ether2	Ethernet	1500	1598	0 bps	0 bps	0	0	0 bps	0 bps
D		ether3	Ethernet	1500	1598	0 bps	0 bps	0	0	0 bps	0 bps
D		ether4	Ethernet	1500	1598	0 bps	0 bps	0	0	0 bps	0 bps
D		pwr-line1	PWR	1500	1598	0 bps	0 bps	0	0	0 bps	0 bps
D		wlan1	Wireless (Atheros AR9	1500	1600	0 bps	0 bps	0	0	0 bps	0 bps

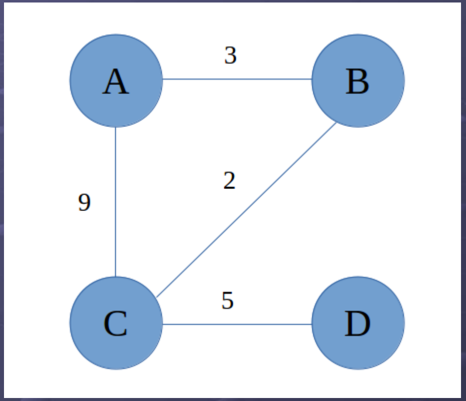
- Q : how do we configure a router HTTP/SSH/Telnet?

RIPv1

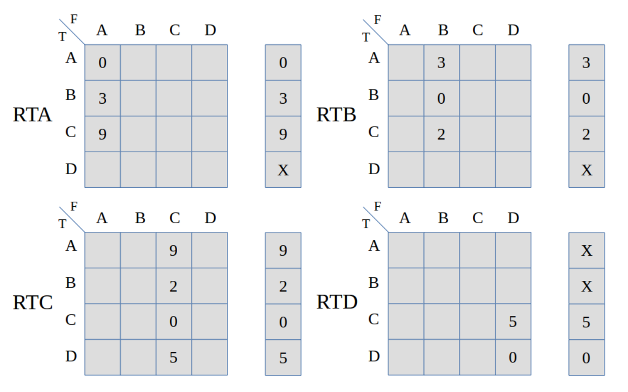
- Routing Information Protocol
  - RFC 1058 : https://datatracker.ietf.org/doc/html/rfc1058
  - RIPv1 created in 1988, was one of the first routing protocols used on the Internet.
  - RIP is a distance vector (DV) routing protocol i.e. routers exchange routing table at regular intervals to calculate the shortest route between two routers.
    - Routing tables exchanged using broadcast packets, approximately every 30 seconds.
    - Distance is measure in terms of the number of routers i.e. hops, between two points in a network.
      - Limited to 15 hops
    - Use the Bellman–Ford algorithm to calculate the best route.

RIPv1

- Consider this network
  - Distance metric is hops.
  - Only showing four routers: A,B,C and D, to simplify discussion.
    - There will be a lot more e.g. from A-C there are nine.
    - Broadcast is limited to local network
  - Could consider metric as representing speed i.e. 1=fast, 10=slow.



RIPv1



RTA

T \ F	A	B	C	D	
A	0				0
B	3				3
C	9				9
D					X

RTB

T \ F	A	B	C	D	
A		3			3
B		0			0
C		2			2
D					X

RTC

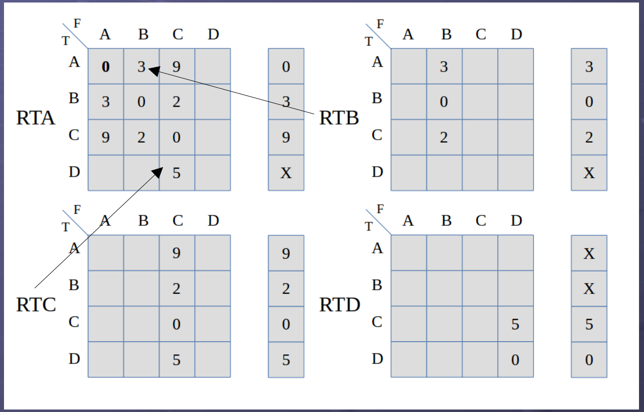
T \ F	A	B	C	D	
A			9		9
B			2		2
C			0		0
D			5		5

RTD

T \ F	A	B	C	D	
A					X
B					X
C				5	5
D				0	0

- T=0 : each router only knows what it can see on its local network.

RIPv1



RTA

T \ F	A	B	C	D	
A	**0**	3	9		0
B	3	0	2		3
C	9	2	0		9
D			5		X

RTB

T \ F	A	B	C	D	
A		3			3
B		0			0
C		2			2
D					X

RTC

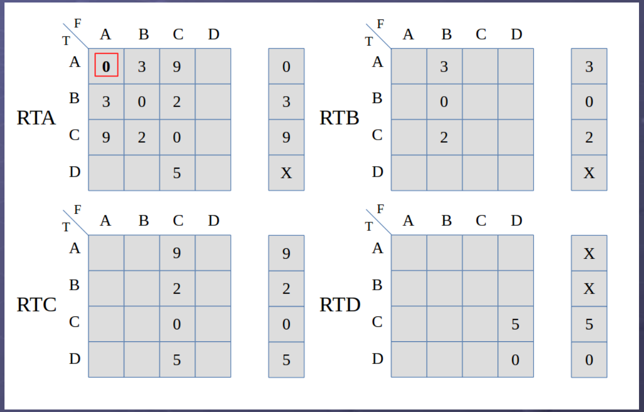
T \ F	A	B	C	D	
A			9		9
B			2		2
C			0		0
D			5		5

RTD

T \ F	A	B	C	D	
A					X
B					X
C				5	5
D				0	0

- T=1 : router A “receives” DV updates from B and C
  - Path to A is minimum, calculate B

RIPv1



RTA

T \ F	A	B	C	D	
A	**0**	3	9		0
B	3	0	2		3
C	9	2	0		9
D			5		X

RTB

T \ F	A	B	C	D	
A		3			3
B		0			0
C		2			2
D					X

RTC

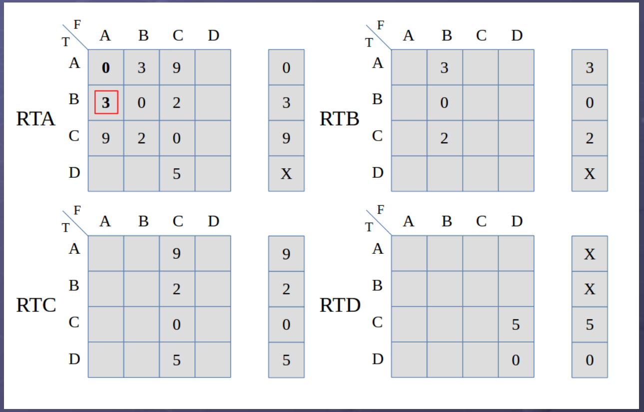
T \ F	A	B	C	D	
A			9		9
B			2		2
C			0		0
D			5		5

RTD

T \ F	A	B	C	D	
A					X
B					X
C				5	5
D				0	0

- T=1 : router A “receives” DV updates from B and C
  - Path to A is minimum, calculate B

RIPv1



RTA

F T	A	B	C	D	
A	**0**	3	9		0
B	**3**	0	2		3
C	9	2	0		9
D			5		X

RTB

F T	A	B	C	D	
A		3			3
B		0			0
C		2			2
D					X

RTC

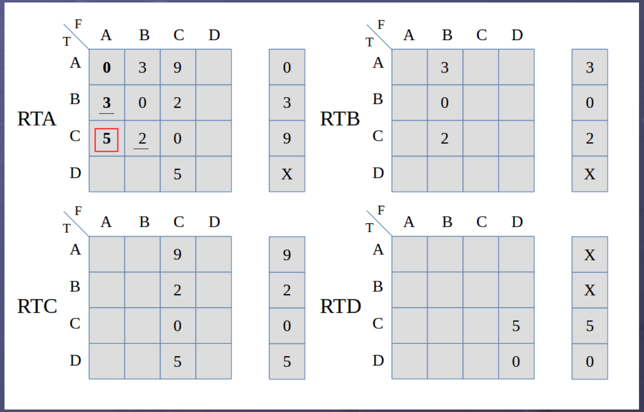
F T	A	B	C	D	
A			9		9
B			2		2
C			0		0
D			5		5

RTD

F T	A	B	C	D	
A					X
B					X
C				5	5
D				0	0

- T=1 : router A “receives” DV updates from B and C
  - Path to B is minimum, calculate C

RIPv1



RTA

F T	A	B	C	D	
A	**0**	3	9		0
B	**3**	0	2		3
C	**5**	2	0		9
D			5		X

RTB

F T	A	B	C	D	
A		3			3
B		0			0
C		2			2
D					X

RTC

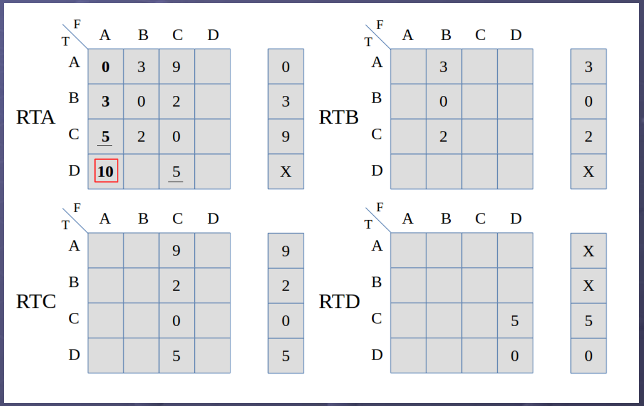
F T	A	B	C	D	
A			9		9
B			2		2
C			0		0
D			5		5

RTD

F T	A	B	C	D	
A					X
B					X
C				5	5
D				0	0

- T=1 : router A “receives” DV updates from B and C
  - Path to C can be improved, calculate D

RIPv1



RTA

F T	A	B	C	D	
A	**0**	3	9		0
B	**3**	0	2		3
C	**5**	2	0		9
D	**10**		5		X

RTB

F T	A	B	C	D	
A		3			3
B		0			0
C		2			2
D					X

RTC

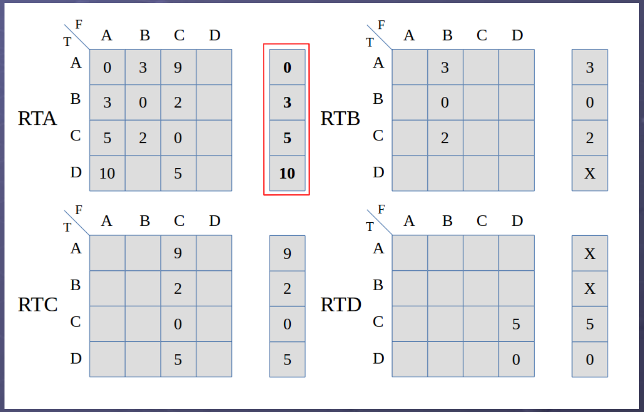
F T	A	B	C	D	
A			9		9
B			2		2
C			0		0
D			5		5

RTD

F T	A	B	C	D	
A					X
B					X
C				5	5
D				0	0

- T=1 : router A “receives” DV updates from B and C
  - Path to D found, update DV

RIPv1



RTA

F T	A	B	C	D	
A	0	3	9		**0**
B	3	0	2		**3**
C	5	2	0		**5**
D	10		5		**10**

RTB

F T	A	B	C	D	
A		3			3
B		0			0
C		2			2
D					X

RTC

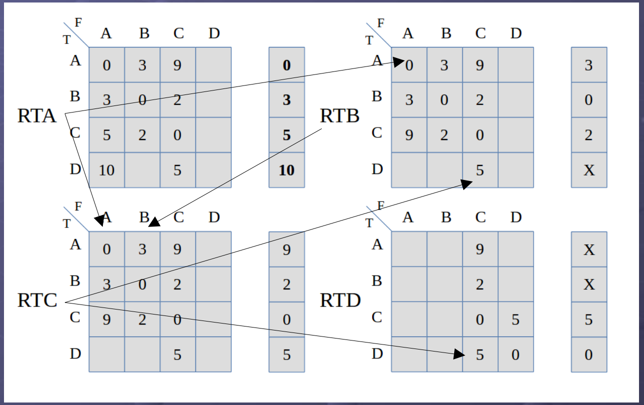
F T	A	B	C	D	
A			9		9
B			2		2
C			0		0
D			5		5

RTD

F T	A	B	C	D	
A					X
B					X
C				5	5
D				0	0

- T=1 : router A “receives” DV updates from B and C
  - Path to D found, update DV

RIPv1



RTA

T\F	A	B	C	D	
A	0	3	9		**0**
B	3	0	2		**3**
C	5	2	0		**5**
D	10		5		**10**

RTB

T\F	A	B	C	D	
A	0	3	9		3
B	3	0	2		0
C	9	2	0		2
D			5		X

RTC

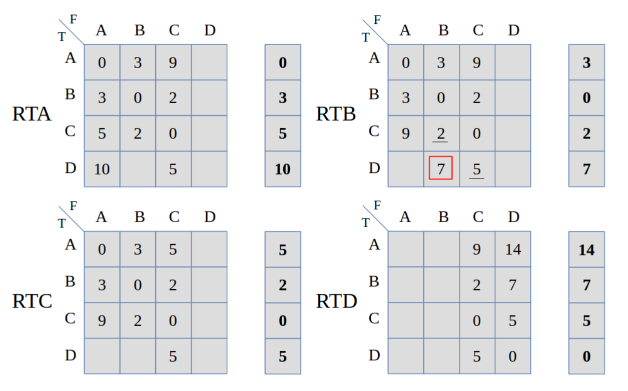
T\F	A	B	C	D	
A	0	3	9		9
B	3	0	2		2
C	9	2	0		0
D			5		5

RTD

T\F	A	B	C	D	
A			9		X
B			2		X
C			0	5	5
D			5	0	0

- T=1 : during this time routers B, C and D also get updates.

RIPv1



RTA

T\F	A	B	C	D	
A	0	3	9		**0**
B	3	0	2		**3**
C	5	2	0		**5**
D	10		5		**10**

RTB

T\F	A	B	C	D	
A	0	3	9		**3**
B	3	0	2		**0**
C	9	2	0		**2**
D		7	5		**7**

RTC

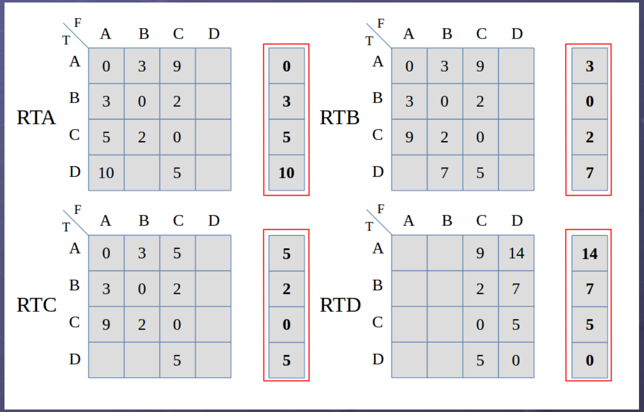
T\F	A	B	C	D	
A	0	3	5		**5**
B	3	0	2		**2**
C	9	2	0		**0**
D			5		**5**

RTD

T\F	A	B	C	D	
A			9	14	**14**
B			2	7	**7**
C			0	5	**5**
D			5	0	**0**

- T=1 : routers B, C and D also update their DV e.g. RTB now knows about a path to D.

RIPv1



RTA

T\F	A	B	C	D	
A	0	3	9		**0**
B	3	0	2		**3**
C	5	2	0		**5**
D	10		5		**10**

RTB

T\F	A	B	C	D	
A	0	3	9		**3**
B	3	0	2		**0**
C	9	2	0		**2**
D		7	5		**7**

RTC

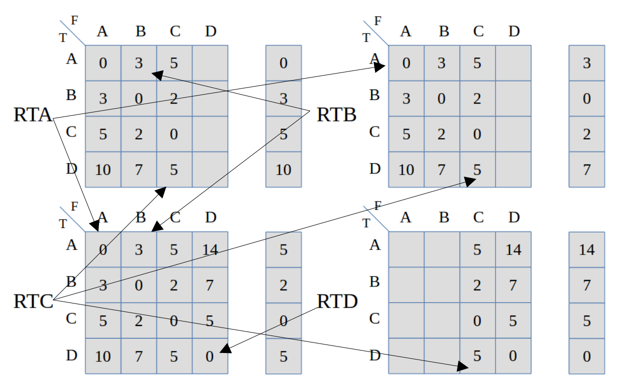
T\F	A	B	C	D	
A	0	3	5		**5**
B	3	0	2		**2**
C	9	2	0		**0**
D			5		**5**

RTD

T\F	A	B	C	D	
A			9	14	**14**
B			2	7	**7**
C			0	5	**5**
D			5	0	**0**

- T=2 : routers A,B,C and D broadcast their DV

RIPv1



RTA

T\F	A	B	C	D	
A	0	3	5		0
B	3	0	2		3
C	5	2	0		5
D	10	7	5		10

RTB

T\F	A	B	C	D	
A	0	3	5		3
B	3	0	2		0
C	5	2	0		2
D	10	7	5		7

RTC

T\F	A	B	C	D	
A	0	3	5	14	5
B	3	0	2	7	2
C	5	2	0	5	0
D	10	7	5	0	5

RTD

T\F	A	B	C	D	
A			5	14	14
B			2	7	7
C			0	5	5
D			5	0	0

- T=2 : routers A,B,C and D broadcast their DV
  - Routes recalculated, process repeated ...

RIPv1

RTA

T \ F	A	B	C	D
A	0	3	5	
B	3	0	2	
C	5	2	0	
D	10	7	5	

0
3
5
10

RTB

T \ F	A	B	C	D
A	0	3	5	
B	3	0	2	
C	5	2	0	
D	10	7	5	

3
0
2
7

RTC

T \ F	A	B	C	D
A	0	3	5	14
B	3	0	2	7
C	5	2	0	5
D	10	7	5	0

5
2
0
5

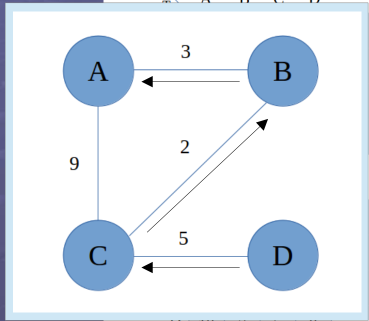
RTD

T \ F	A	B	C	D
A			5	14
B			2	7
C			0	5
D			5	0

14
7
5
0

- T=2 : routers A,B,C and D broadcast their DV
  - Routes recalculated, process repeated ...

RIPv1



0
3
5
10

RTB

T \ F	A	B	C	D
A	0	3	5	
B	3	0	2	
C	5	2	0	
D	10	7	5	

3
0
2
7

5
2
0
5

RTD

T \ F	A	B	C	D
A			5	10
B			2	7
C			0	5
D			5	0

10
7
5
0

- T=2 : routers A,B,C and D broadcast their DV
  - Allows routers to discover “best” routes

RIPv1

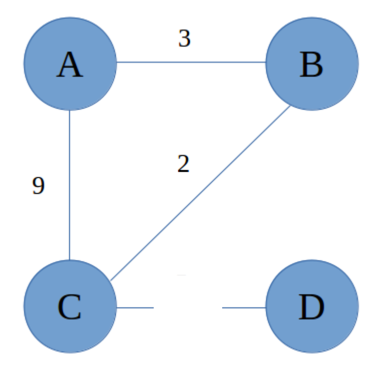
- Advantages
  - Simple, lightweight, it works, “fine” for small networks.
- Disadvantages
  - For larger networks limitations become an issue e.g. limited to 15 hops, no routing metrics, passwords, does not support variable length subnet masks ...
  - Distance vectors transferred using broadcast packets i.e. host not involved in routing will also have to process these packets.
  - Requests for updates made by router every 30 seconds.
    - When you have a lot of routers, that’s a lot of traffic.
  - Only remembers the lowest cost route i.e. no backup routes stored.

RIPv2

- Routing Information Protocol
  - RFC 1723 : https://datatracker.ietf.org/doc/html/rfc1723
  - RIPv2 created in 1994, correcting some of the previous issues with RIPv1 e.g. supports CIDR, MD5 encryption / passwords.
  - Routing tables exchanged using multicast packets i.e. only host involved with routing with process these packets.
    - Assigned multicast address: 224.0.0.9
  - For backwards comparability still limited to 15 hops, also suffers from slow convergence and count to infinity problem.

RIPv2

- Consider this network
  - Path from C to D is lost
  - Router C detects that this connection is lost, therefore, deletes this entry and updates its DV ...



RIPv2

RTA

T\F	A	B	C	D	
A	0	3	5		0
B	3	0	2		3
C	5	2	0		5
D	10	7	5		10

RTB

T\F	A	B	C	D	
A	0	3	5		3
B	3	0	2		0
C	5	2	0		2
D	10	7	5		7

RTC

T\F	A	B	C	D	
A	0	3	5	14	5
B	3	0	2	7	2
C	5	2	0	5	0
D	10	7	5	0	5

RTD

T\F	A	B	C	D	
A			5	10	10
B			2	7	7
C			0	5	5
D			5	0	0

- DV before path from C to D is lost.

RIPv2

RTA

T\F	A	B	C	D	
A	0	3	5		0
B	3	0	2		3
C	5	2	0		5
D	10	7	5		10

RTB

T\F	A	B	C	D	
A	0	3	5		3
B	3	0	2		0
C	5	2	0		2
D	10	7	5		7

RTC

T\F	A	B	C	D	
A	0	3	5		5
B	3	0	2		2
C	5	2	0		0
D	10	7	**9**		**9**

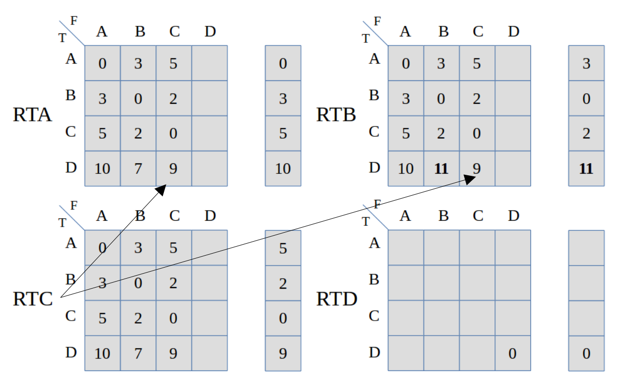
RTD

T\F	A	B	C	D	
A					
B					
C					
D				0	0

- Router C detects that path to D is lost and updates its routing table

---

RIPv2



RTA

T\F	A	B	C	D	
A	0	3	5		0
B	3	0	2		3
C	5	2	0		5
D	10	7	9		10

RTB

T\F	A	B	C	D	
A	0	3	5		3
B	3	0	2		0
C	5	2	0		2
D	10	**11**	9		**11**

RTC

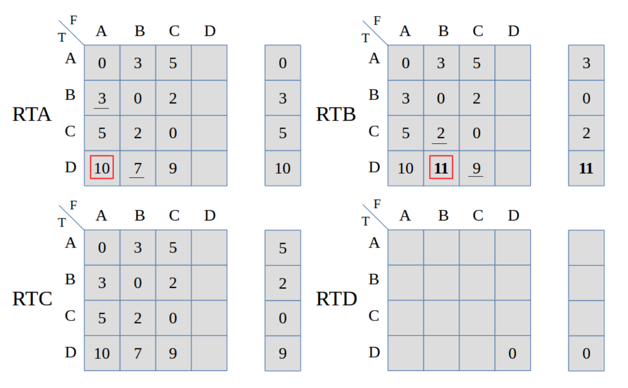
T\F	A	B	C	D	
A	0	3	5		5
B	3	0	2		2
C	5	2	0		0
D	10	7	9		9

RTD

T\F	A	B	C	D	
A					
B					
C					
D				0	0

- Router C broadcasts it DV to A and B
  - Routers A and B update their DV

RIPv2



RTA

T/F	A	B	C	D	
A	0	3	5		0
B	3	0	2		3
C	5	2	0		5
D	10	7	9		10

RTB

T/F	A	B	C	D	
A	0	3	5		3
B	3	0	2		0
C	5	2	0		2
D	10	11	9		11

RTC

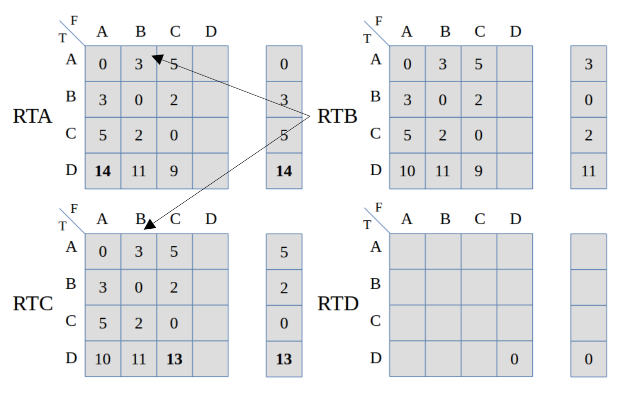
T/F	A	B	C	D	
A	0	3	5		5
B	3	0	2		2
C	5	2	0		0
D	10	7	9		9

RTD

T/F	A	B	C	D	
A					
B					
C					
D				0	0

- Router C broadcasts it DV to A and B
  - Routers A and B update their DV

RIPv2



RTA

T/F	A	B	C	D	
A	0	3	5		0
B	3	0	2		3
C	5	2	0		5
D	**14**	11	9		**14**

RTB

T/F	A	B	C	D	
A	0	3	5		3
B	3	0	2		0
C	5	2	0		2
D	10	11	9		11

RTC

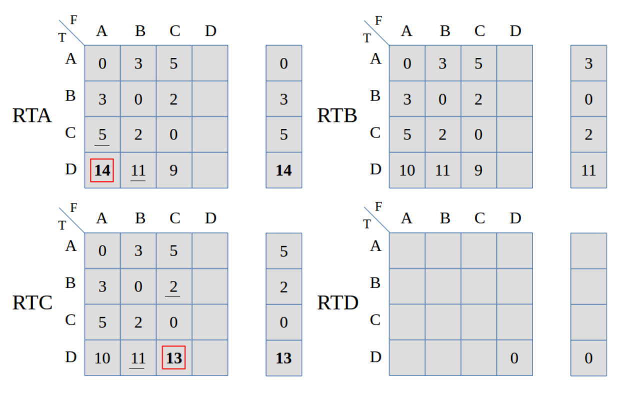
T/F	A	B	C	D	
A	0	3	5		5
B	3	0	2		2
C	5	2	0		0
D	10	11	**13**		**13**

RTD

T/F	A	B	C	D	
A					
B					
C					
D				0	0

- Router B broadcasts its DV, A and C update their DV, then broadcast their updated DV

RIPv2



RTA

T/F	A	B	C	D	
A	0	3	5		0
B	3	0	2		3
C	5	2	0		5
D	**14**	11	9		**14**

RTB

T/F	A	B	C	D	
A	0	3	5		3
B	3	0	2		0
C	5	2	0		2
D	10	11	9		11

RTC

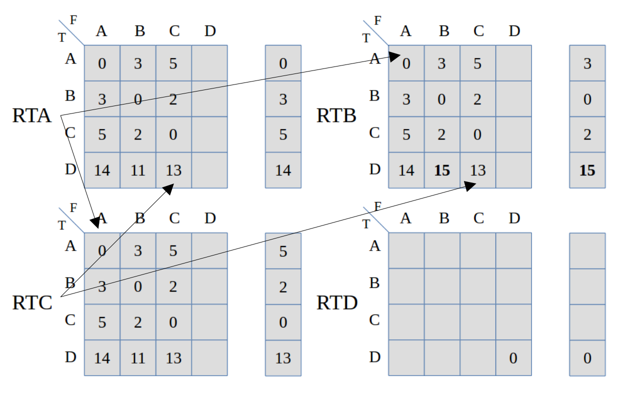
T/F	A	B	C	D	
A	0	3	5		5
B	3	0	2		2
C	5	2	0		0
D	10	11	**13**		**13**

RTD

T/F	A	B	C	D	
A					
B					
C					
D				0	0

- Router B broadcasts its DV, A and C update their DV, then broadcast their updated DV

RIPv2



RTA

T/F	A	B	C	D	
A	0	3	5		0
B	3	0	2		3
C	5	2	0		5
D	14	11	13		14

RTB

T/F	A	B	C	D	
A	0	3	5		3
B	3	0	2		0
C	5	2	0		2
D	14	**15**	13		**15**

RTC

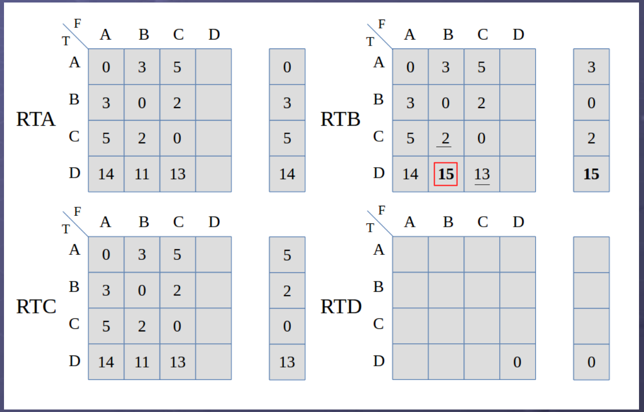
T/F	A	B	C	D	
A	0	3	5		5
B	3	0	2		2
C	5	2	0		0
D	14	11	13		13

RTD

T/F	A	B	C	D	
A					
B					
C					
D				0	0

- Router B updates …
  - Routes over 15 are removed, otherwise will count to infinity.

RIPv2



RTA

T \ F	A	B	C	D	
A	0	3	5		0
B	3	0	2		3
C	5	2	0		5
D	14	11	13		14

RTB

T \ F	A	B	C	D	
A	0	3	5		3
B	3	0	2		0
C	5	$\underline{2}$	0		2
D	14	$\boxed{\textbf{15}}$	$\underline{13}$		**15**

RTC

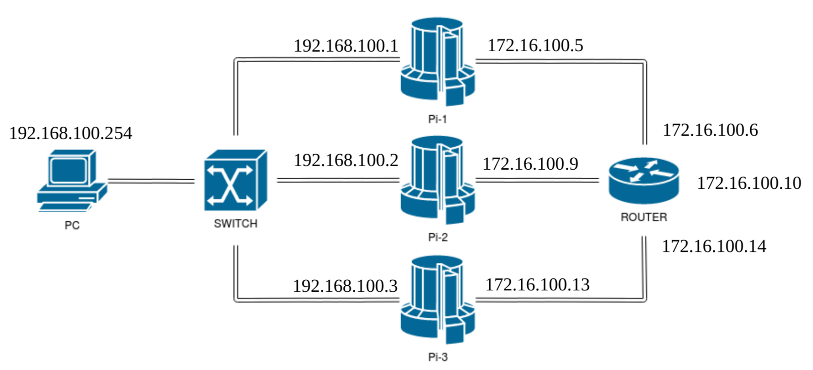
T \ F	A	B	C	D	
A	0	3	5		5
B	3	0	2		2
C	5	2	0		0
D	14	11	13		13

RTD

T \ F	A	B	C	D	
A					
B					
C					
D				0	0

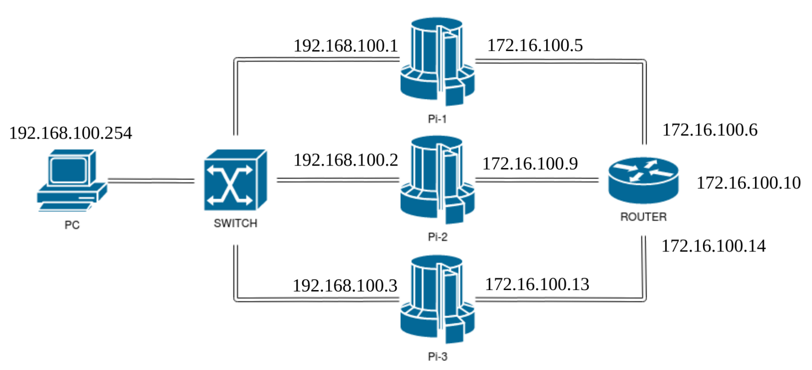
- Router B updates …
  - Routes over 15 are removed, otherwise will count to infinity.

Test network



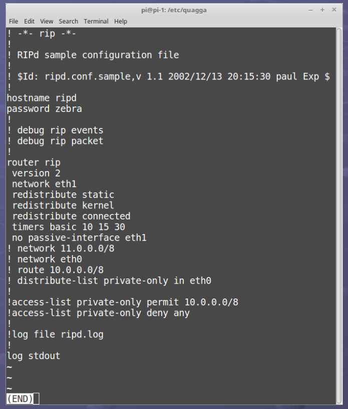
- Four hosts, four subnets, one switch and one router.

Quick Quizzz

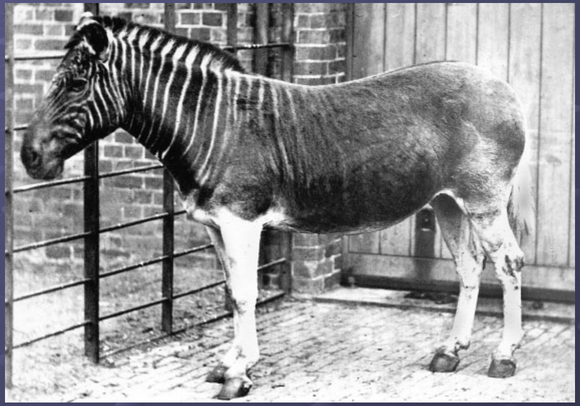


- Q : How many collision domains and Broadcast domains are there in this network?

Quagga

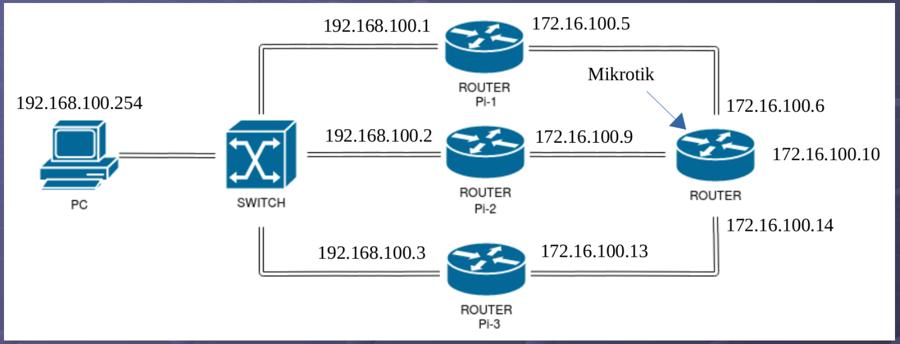


```
! -*- rip -*-
!
! RIPd sample configuration file
!
! $Id: ripd.conf.sample,v 1.1 2002/12/13 20:15:30 paul Exp $
!
hostname ripd
password zebra
!
! debug rip events
! debug rip packet
!
router rip
 version 2
 network eth1
 redistribute static
 redistribute kernel
 redistribute connected
 timers basic 10 15 30
 no passive-interface eth1
! network 11.0.0.0/8
! network eth0
! route 10.0.0.0/8
! distribute-list private-only in eth0
!
!access-list private-only permit 10.0.0.0/8
!access-list private-only deny any
!
!log file ripd.log
!
log stdout
~
~
~
(END)
```



- Quagga : turn a Raspberry Pi into a route
  - A fork from Zebra

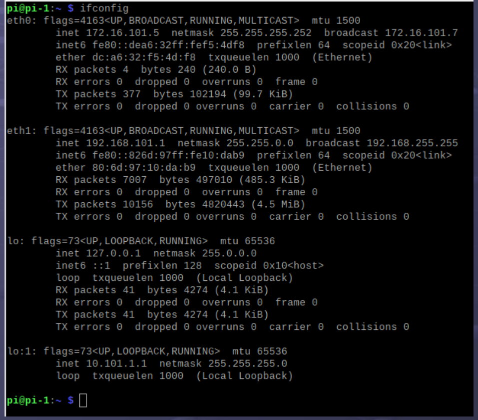
Quagga



- Seven subnets, four routers, one switch and one host (depending on point of view)
  - How many broadcast domains are there now?

Demo : Test network

- Subnets (X=Desk)
  - 192.168.0.0/16
  - 172.16.X.4/30
  - 172.16.X.8/30
  - 172.16.X.12/30
  - 10.X.1.0/24
  - 10.X.2.0/24
  - 10.X.3.0/24



```
pi@pi-1:~ $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.101.5  netmask 255.255.255.252  broadcast 172.16.101.7
        inet6 fe80::dea6:32ff:fef5:4df8  prefixlen 64  scopeid 0x20<link>
        ether dc:a6:32:f5:4d:f8  txqueuelen 1000  (Ethernet)
        RX packets 4  bytes 240 (240.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 377  bytes 102194 (99.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.101.1  netmask 255.255.0.0  broadcast 192.168.255.255
        inet6 fe80::826d:97ff:fe10:dab9  prefixlen 64  scopeid 0x20<link>
        ether 80:6d:97:10:da:b9  txqueuelen 1000  (Ethernet)
        RX packets 7007  bytes 497010 (485.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 10156  bytes 4820443 (4.5 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 41  bytes 4274 (4.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 41  bytes 4274 (4.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo:1: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 10.101.1.1  netmask 255.255.255.0
        loop  txqueuelen 1000  (Local Loopback)

pi@pi-1:~ $ 
```

OSPF

- Open Shortest Path First
  - OSPFv1 (1989) - RFC 1131 :
    - https://datatracker.ietf.org/doc/html/rfc1131
  - OSPFv2 (1998) - RFC 2328 :
    - https://datatracker.ietf.org/doc/html/rfc2328
  - Like RIP this is an Interior Gateway Protocol (IGP) used to transfer routing table information between routers within an Autonomous System (AS) e.g. a LAN.
  - Uses the Link State Routing algorithm (LSR) i.e. a dynamic routing protocol that can calculate the fastest route to a destination.
    - Routing decisions now also based on the speed of the communications path / routers (bps)
  - In general has replaced RIP.

OSPF

- Advantages
  - No limitations on hop count
  - Supports Variable Length Subnet Masks (VLSM)
  - Multicast used to communicating link-state updates
  - Quicker convergence than RIP, supports authentication ...
- Disadvantages
  - Calculating link state is processor intensive
  - Maintaining routing information regarding link state significantly increases memory requirements
  - Not as easy to learn / understand as some other protocols e.g. RIP :)
- Nice video : https://www.youtube.com/watch?v=kfvJ8QVJscc

BGP

- Border Gateway Protocol (BGP) – RFC 1105 :
    - https://datatracker.ietf.org/doc/html/rfc1105
  - Exterior Gateway Protocol (EGP) designed to exchange routing information amongst autonomous systems (AS) on the Internet.
    - Consider ASs (networks) like your tier-1 service providers, these will have 1000s of routers e.g. AT&T, BT ...
  - Edge routers (eBGP) on the boundary of autonomous systems exchange information with another edge routers.
    - Deciding what the best route is between ASs is “tricky” when your negating between untrusted neighbours.
    - Different ASs will have different views of what the best path is.
      - Cost, Capacity, Speed. Here be dragons :)
  - Nice video: https://www.youtube.com/watch?v=-wMU8vmfaYo (DNS BGP)

Summary

- Fragmentation allows larger packets to be transferred over networks that have a smaller MTU.
  - IPv4 : both hosts and routers can fragment
  - IPv6 : only hosts can fragment.
  - Normally discouraged as reassembly is computational expensive and inefficient. Also has security issues e.g. DoS attacks.
- Router configuration can be tricky.
- Routing protocols are a module in their own right. This was only an introduction, I leave the background reading to you :)

Summary

- However, we still have some unanswered questions:
  - How are the raw data bits transmitted across a cable?
  - Do we need other protocols to do this?
    - Logical addressing vs Physical addressing
  - What protocols have we already used and not looked at yet?